

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (original) A method for decoding data, said method comprising iterations with some steps (SISO1, SISO2) using windows (WID) of input data, characterized in that the method comprises, for a current window (WID) of a step (SISO1, SISO2) within an iteration the steps of:  
Performing a forward recursion, wherein said forward recursion is initialized with a forward state metric vector ( $\alpha$ ) from the upper stake (STK) of a previous window of the same step (SISO1, SISO2) of a previous iteration, a window (WID) comprising a lower and an upper stake (STK), and  
Performing a backward recursion, wherein said backward recursion is initialized with a backward state metric vector ( $\alpha$ ) from the lower stake (STK) of a next window of the same step (SISO1, SISO2) of a previous iteration.
2. (original) A method as claimed in claim 1, characterized in that the forward state metric vector ( $\alpha$ ) computed last is stored in an upper stake of said current window (WID) during the forward recursion, and the backward state metric vector ( $\beta$ ) computed last is stored in the lower stake (STK) of said current window (WID) during the backward recursion.
3. (original) A method as claimed in claim 1, characterized in that all the windows (WID) of a step (SISO) are processed in parallel.
4. (original) A decoder for decoding data, said decoding comprising iterations with some steps (SISO1, SISO2) using windows (WID) of input data, characterized in

that it comprises computation units (CMP) for performing, for a current window (WID) of a step (SISO1, SISO2) within an iteration:

A forward recursion, wherein said forward recursion is initialized with a forward state metric vector ( $\alpha$ ) from the upper stake (STK) of a previous window of the same step (SISO1, SISO2) of a previous iteration, a window (WID) comprising a lower and an upper stake (STK), and

A backward recursion, wherein said backward recursion is initialized with a backward state metric vector ( $\beta$ ) from the lower stake (STK) of a next window of the same step (SISO1, SISO2) of a previous iteration.

5. (original) A receiver adapted to receive input data, said input data being processed by the decoder as claimed in claim 4.
6. (original) A computer program product for a receiver, comprising a set of instructions which, when loaded into said receiver, causes the receiver to carry out the method as claimed in claims 1 to 3.
7. (original) A computer program product for a computer, comprising a set of instructions which, when loaded into said computer, causes the computer to carry out the method as claimed in claims 1 to 3.
8. (new) A decoder for decoding data, as claimed in claim 4, said decoding characterized in that the forward state metric vector ( $\alpha$ ) computed last is stored in an upper stake of said current window during the forward recursion, and the backward state metric vector ( $\beta$ ) computed last is stored in the lower stake of said current window during the backward recursion.
9. (new) A decoder for decoding data, as claimed in claim 4, said decoding characterized in that all the windows of a step are processed in parallel.

10. (new) A method as claimed in claim 1, characterized in that the backward recursion is initialized with a metrics vector computed by a termination generator, wherein the metric vector is a function of tail bits, and is processed.
11. (new) A decoder for decoding data, as claimed in claim 4, said decoding characterized in that the backward recursion is initialized with a metrics vector computed by a termination generator, wherein the metric vector is a function of tail bits, and is processed.